

Задание - расчет прибыльности

Расчет себестоимости по FIFO – метод учета по цене первой поступившей или изготовленной партии.

Пример:

Приемка (закупка) от 01.01.17: 1 телефон по цене 1000 р. каждый

Приемка (закупка) от 01.02.17 – 2 телефона по цене 2000 р. каждый

Отгрузка (продажа) от 01.03.17 – 2 телефона по цене 5000 р. каждый

Расчет:

Сумма себестоимости 2 проданных штук = $1000 + 2000 = 3000$

Себестоимость единицы = $3000/2 = 1500$

Прибыль = $2*5000 - 3000 = 7000$

Постановка задачи

Необходимо реализовать консольное приложение на Java по расчету прибыльности. Управление программой осуществляется посредством ввода команд, подаваемых в поток стандартного ввода. Команды разделены переносом строки. Команды имеют вид: <Наименование команды><параметры, разделённые пробелами>. Команды должны обрабатываться по одной. Результат обработки команды - либо ОК, либо ERROR, либо значение, - выводятся в стандартный поток вывода.

Список команд

- 1) NEWPRODUCT <name> - Создать товар - на вход подается уникальное наименование товара
- 2) PURCHASE <name> <amount> <price> <date> - Закупить товар - на вход подается наименование товара, кол-во закупленного товара, цена единицы товара и дата закупки
- 3) DEMAND <name> <amount> <price> <date> - Продать товар - на вход подается наименование товара, кол-во проданного товара, цена единицы товара и дата продажи
- 4) SALESREPORT <name> <date> - Рассчитать прибыльность - на вход подается наименование товара и дата. Результат - прибыль на указанную дату - выводится в стандартный поток вывода

Пример потока команд

NEWPRODUCT iphone

NEWPRODUCT iphone

PURCHASE iphone 1 1000 01.01.2017

PURCHASE iphone 2 2000 01.02.2017

DEMAND iphone 2 5000 01.03.2017

SALESREPORT iphone 02.03.2017

программа должна вывести:

OK

ERROR

OK

OK

OK

7000

Особенности объектной модели

- нельзя завести 2 раза товар с одинаковым наименованием
- перед тем как покупать/продавать товар, его необходимо создать
- цены товара должны быть больше нуля

Функциональные требования

- команды должны валидировать входные параметры
- программа не должна "упасть" на некорректных данных

Особенности реализации задания

Необходимо реализовать сборку с помощью инструмента Maven.

Покрытие тестами является обязательным.

Задание должно быть выложено в публичный репозиторий на github. В файле README необходимо указать параметры сборки, если они отличаются от стандартных (mvn clean package).

NOTE 1

Данные можно хранить в памяти (например, в виде списков), либо использовать для хранения PostgreSQL / H2. Во втором случае README файл необходимо дополнить данными, необходимыми для развертывания БД.

NOTE 2

Если вы имели опыт работы с REST, данное задание можно выполнить в виде API, работу с данными (отправка и получение) организовать в формате JSON. Использование Spring **запрещается**, т.к. сильно упрощает задачу и не дает возможности увидеть навыки кандидата.

К реализованному API составить небольшую документацию (в файле README) - список возможных команд с передаваемыми в них параметрами и возможными ответами на запрос. Файл с документацией может быть в формате Markdown, либо с использованием инструментов для документирования АПИ(API Blueprint, Swagger и подобные)